

Software-Projekt I

Prof. Dr. Rainer Koschke

Arbeitsgruppe Softwaretechnik
Fachbereich Mathematik und Informatik
Universität Bremen

Sommersemester 2013

Vorbemerkungen I

1 Vorbemerkungen

- Ziele und Inhalt
- Übungsbetrieb
- Angestrebte Resultate
- Aktivitäten der Software-Entwicklung
- Projektablauf
- Scheinbedingungen
- Anmeldung
- Aufgabe im SWP II
- Ressourcen





Die zehn teuflischen SWP-Gebote

- ① Du sollst nicht in die Vorlesung gehen.
- ② Wenn (1) doch geschieht, dann sollst du nur spielen und surfen.
- ③ Du sollst die Vorlesung nicht vor- und nachbereiten und keine Notizen machen.
- ④ Du sollst keinesfalls Fragen während der Vorlesung stellen.
- ⑤ Du sollst nicht in die Übungen gehen.
- ⑥ Du sollst niemals die Übungszettel vor der Prüfung anschauen.
- ⑦ Du sollst die Musterklausur erst nach der Klausur anschauen.
- ⑧ Wenn (7) doch geschieht, sollst du keinesfalls die Musterklausur vor der Klausur lösen.
- ⑨ Du sollst frühestens drei Tage vor der Prüfung anfangen zu lernen.
- ⑩ Du sollst nach dem Nichtbestehen nicht in die Klausureinsicht gehen.

Ziele der Vorlesung

Primäre Ziele:

- Rüstzeug für erfolgreiche Durchführung eures Software-Projekts vermitteln
 - Modell für zukünftige ähnliche Projekte bieten
- Praktikum und Vorlesung bilden eine Einheit

Primäre Ziele sind **nicht**:

- Vollständige Darstellung aller Themengebiete der Softwaretechnik¹
- Vermittlung von spezifischen Kenntnissen für die Entwicklung des Anwendungssystems X

¹Wird im Hauptstudium nachgereicht.

Szenario dieses Software-Projekts

- Erstellung eines großen Softwaresystems
 - ▶ mehrere Mitarbeiter über einen langen Zeitraum
- für einen realen Auftraggeber
 - ▶ Individualsoftware
- im Rahmen eines Projekts
 - ▶ einmalige Zielverfolgung
 - ▶ dennoch: Software soll wartbar sein

Stufen zum Erfolg

- zu votierende Übungsaufgaben mit Tutorium
- in Vorlesung und Tutorium integrierter Technikkurs
- Tools, Frameworks und Beispiele im Tutorium
- Mini-Projekt begleitend zu SWP-I:
 - ▶ Abgabe von Dokumenten während der Vorlesungszeit
 - ▶ Implementierung und Test im einwöchigen Blockkurs in vorlesungsfreier Zeit (Durchstich für 2-3 Anwendungsfälle)
- im WS: SWP-II-Projekt: Ausbau der Ergebnisse aus SWP-I

Angestrebte Resultate

- Softwaretechnik ist nicht nur Programmieren.

Angestrebte Resultate

- Softwaretechnik ist nicht nur Programmieren.
- Softwaretechnik ist auch Programmieren.

Angestrebte Resultate

- Softwaretechnik ist nicht nur Programmieren.
- Softwaretechnik ist auch Programmieren.
- Software-Entwicklung produziert Dokumente.
- Der Quellcode ist **ein** Dokument **unter vielen**.

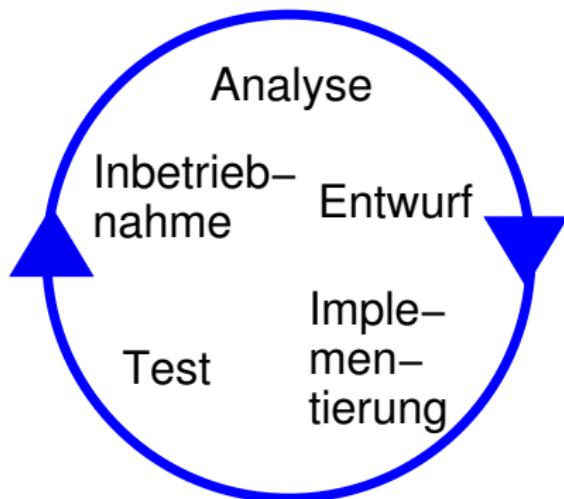
Angestrebte Resultate

- Softwaretechnik ist nicht nur Programmieren.
- Softwaretechnik ist auch Programmieren.
- Software-Entwicklung produziert Dokumente.
- Der Quellcode ist **ein** Dokument **unter vielen**.

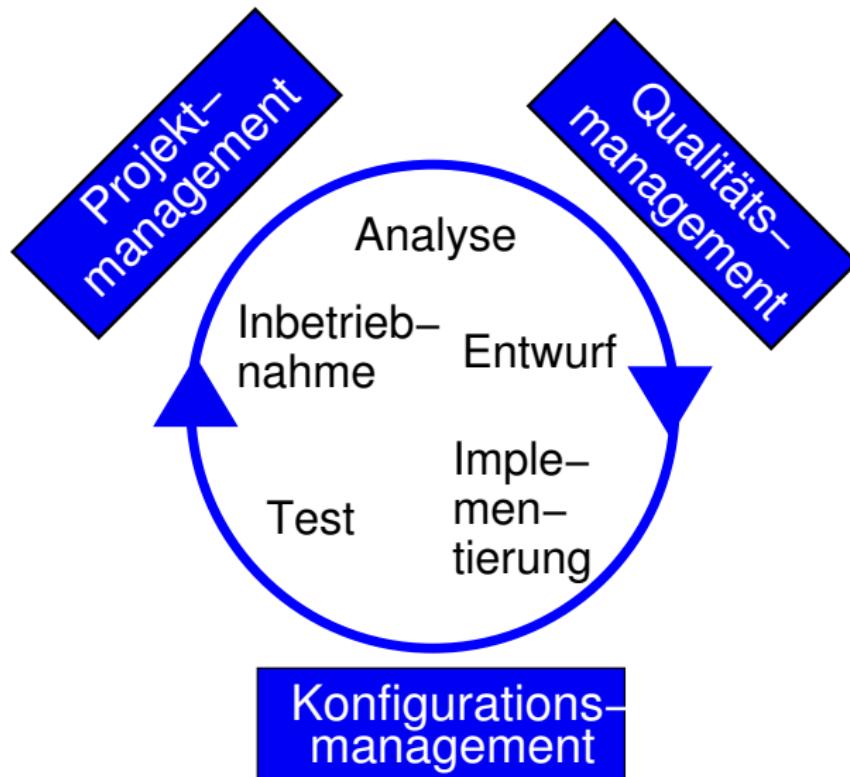
Ihr sollt nicht nur ein richtiges System bauen.

Ihr sollt ein System auch richtig bauen.

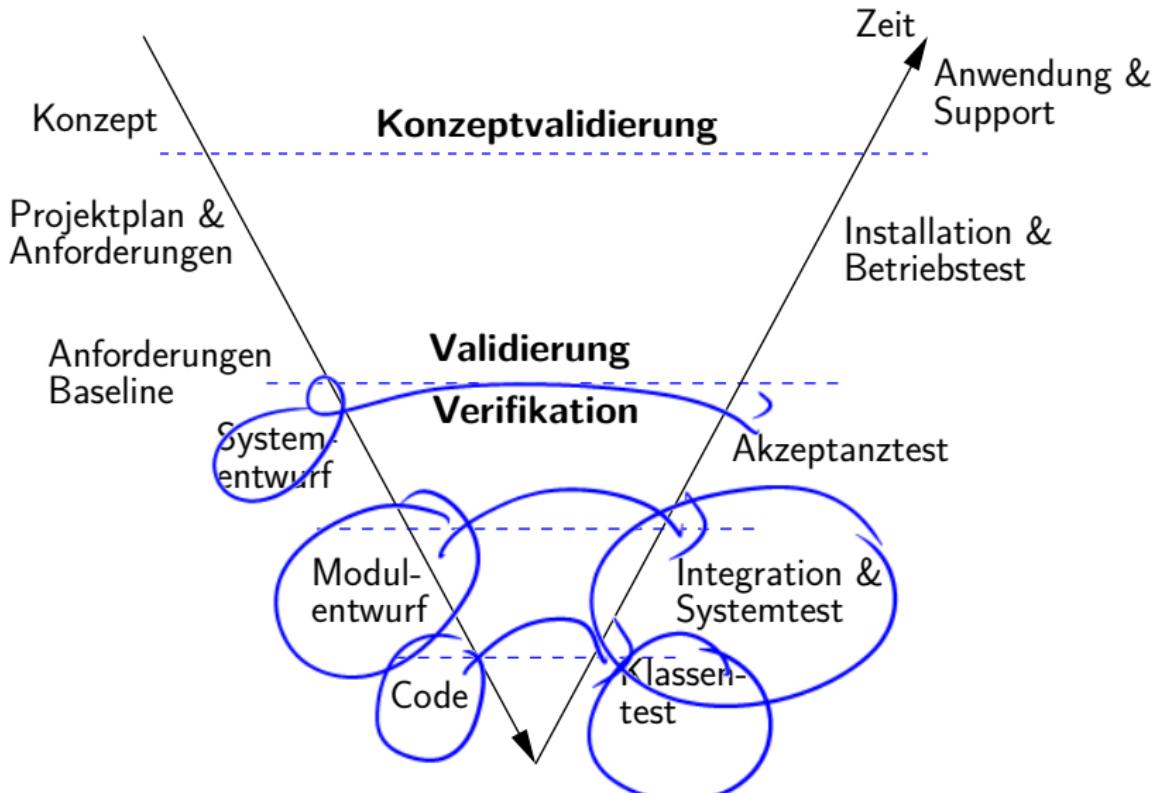
Aktivitäten bei der Softwareentwicklung



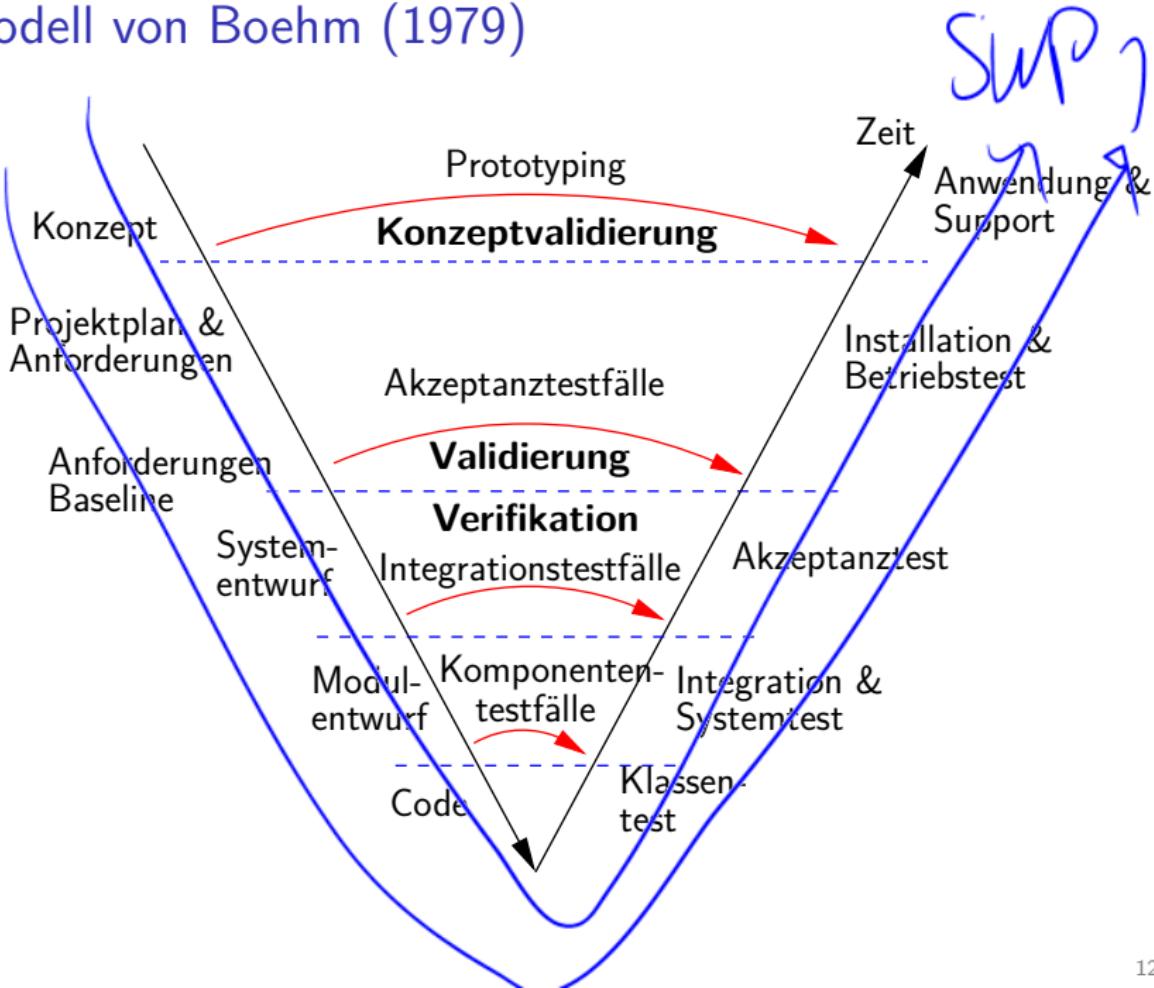
Aktivitäten bei der Softwareentwicklung



V-Modell von Boehm (1979)



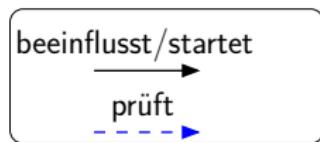
V-Modell von Boehm (1979)



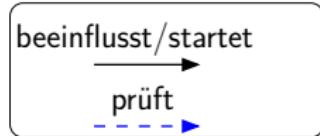
Infrastruktur einrichten, Teambuilding



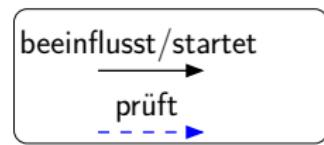
- ─ Infrastruktur einrichten, Teambuilding
- Initiale Projektplanung

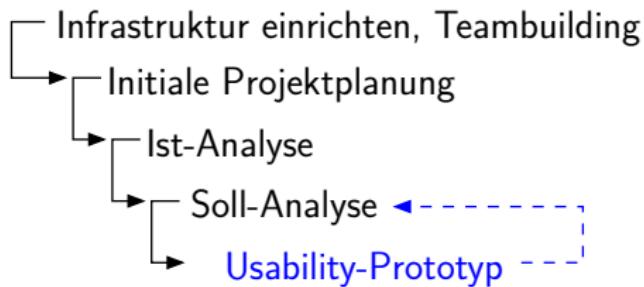


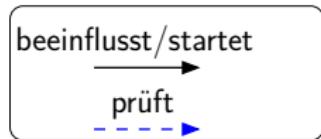
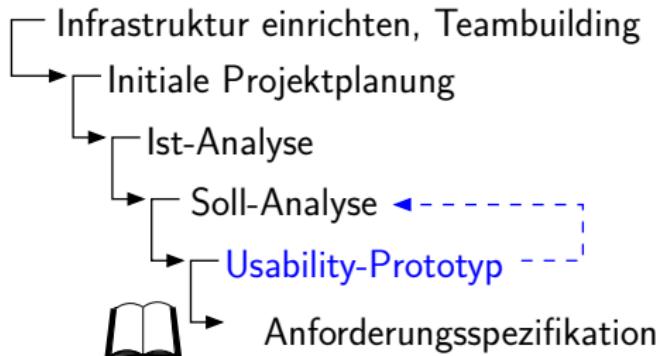
- └─ Infrastruktur einrichten, Teambuilding
- └─ Initiale Projektplanung
 - └─ Ist-Analyse

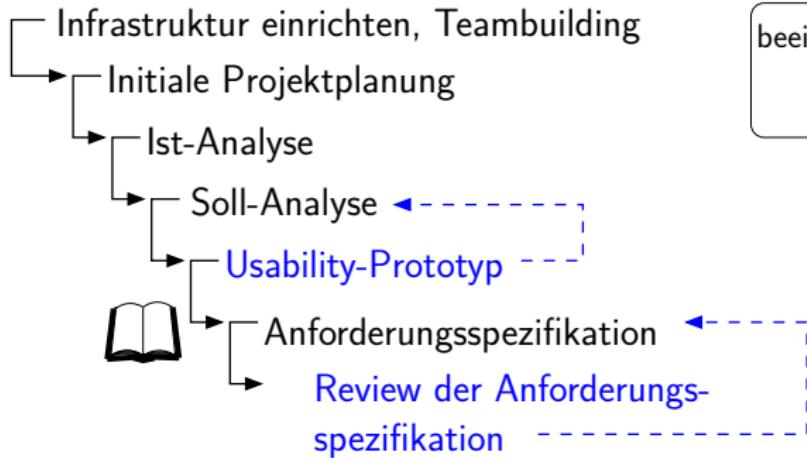


- └─ Infrastruktur einrichten, Teambuilding
- └─ Initiale Projektplanung
 - └─ Ist-Analyse
 - └─ Soll-Analyse

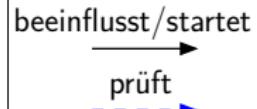








Infrastruktur einrichten, Teambuilding



Initiale Projektplanung

Ist-Analyse

Soll-Analyse

Usability-Prototyp



Anforderungsspezifikation



Review der Anforderungs-
spezifikation

Handbuch
schreiben

Infrastruktur einrichten, Teambuilding

Initiale Projektplanung

Ist-Analyse

Soll-Analyse

Usability-Prototyp



Anforderungsspezifikation

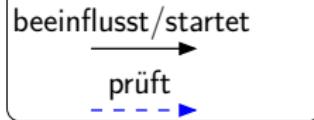
Review der Anforderungs-
spezifikation



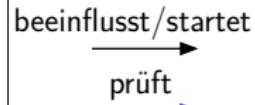
Handbuch
schreiben



Architekturentwurf



Infrastruktur einrichten, Teambuilding



Initiale Projektplanung

Ist-Analyse

Soll-Analyse

Usability-Prototyp



Anforderungsspezifikation

Review der Anforderungs-
spezifikation



Handbuch
schreiben



Architekturentwurf
Technisches
Prototyping

Infrastruktur einrichten, Teambuilding

Initiale Projektplanung

Ist-Analyse

Soll-Analyse

Usability-Prototyp



Anforderungsspezifikation

Review der Anforderungs-
spezifikation



Handbuch
schreiben

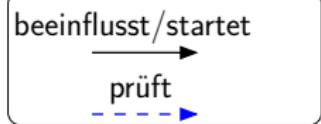


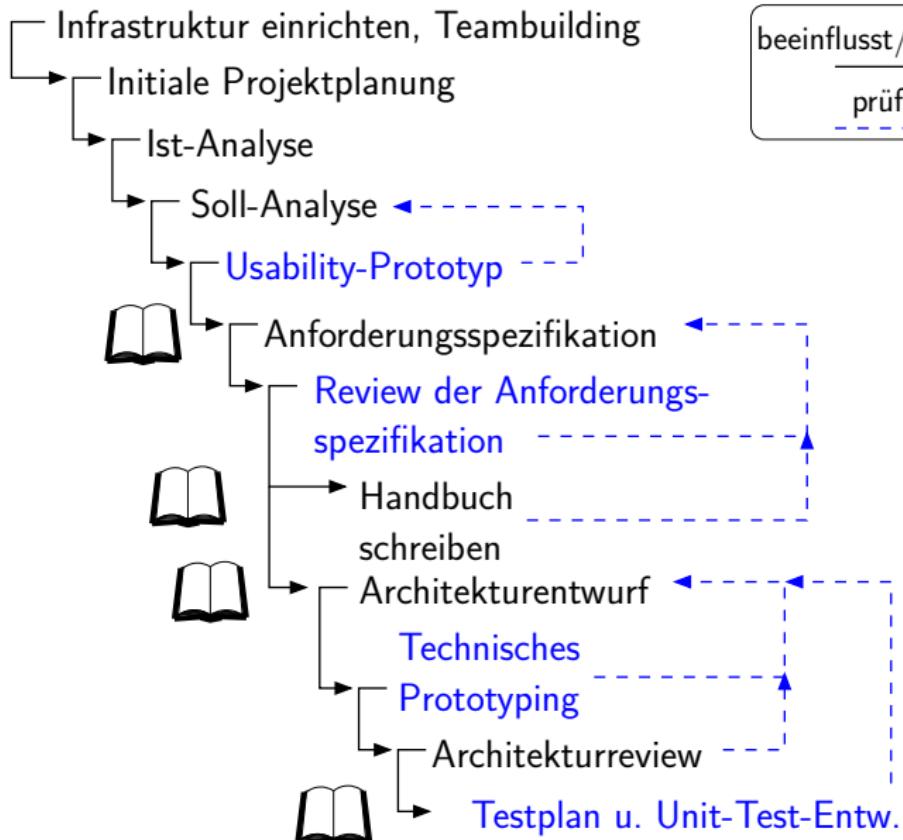
Architekturentwurf

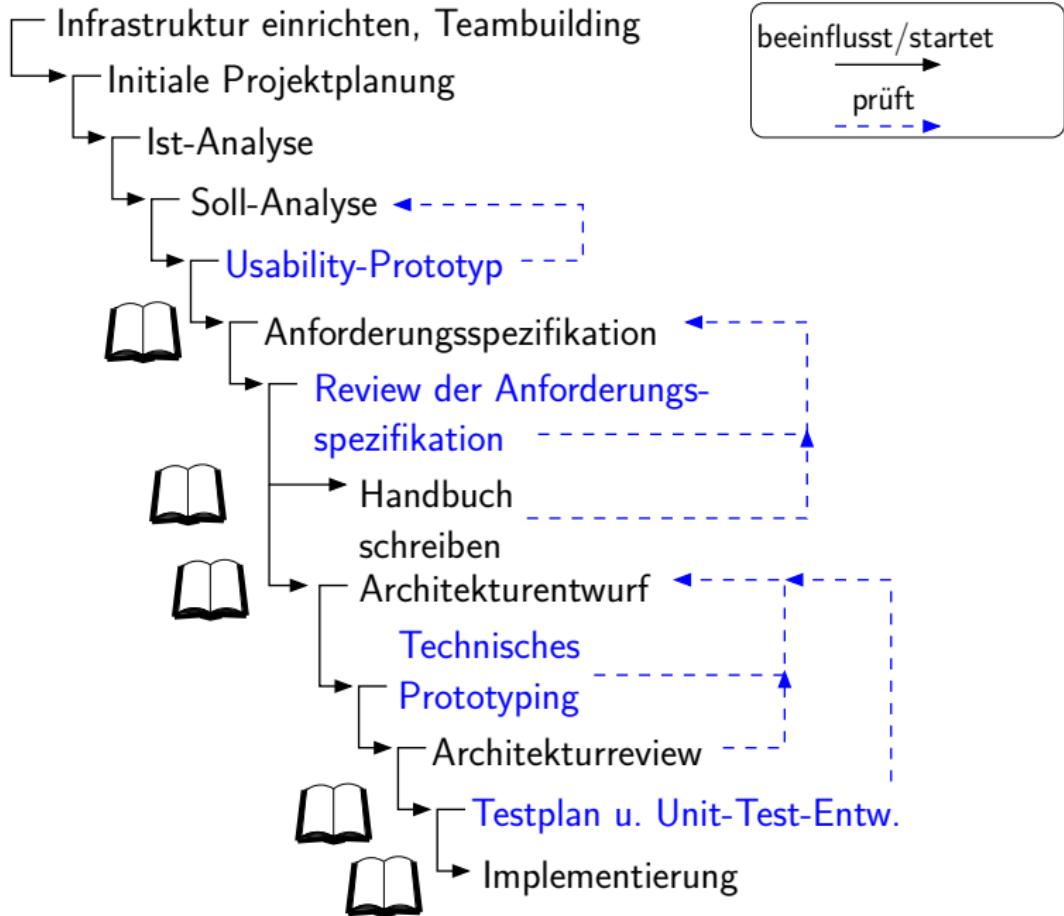
Technisches

Prototyping

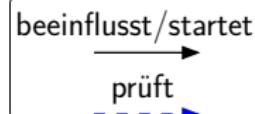
Architekturreview







Infrastruktur einrichten, Teambuilding



Initiale Projektplanung

Ist-Analyse

Soll-Analyse

Usability-Prototyp



Anforderungsspezifikation



Review der Anforderungs-
spezifikation

Handbuch
schreiben



Architekturentwurf

Technisches

Prototyping



Architekturreview

Testplan u. Unit-Test-Entw.



Implementierung

Code-Inspektion

Unit-/Integrationstests

Infrastruktur einrichten, Teambuilding

beeinflusst/startet

Initiale Projektplanung

prüft

Ist-Analyse

Soll-Analyse

Usability-Prototyp



Anforderungsspezifikation



Review der Anforderungs-
spezifikation



Handbuch
schreiben



Testprotokoll

Architekturentwurf

Systemtests

Technisches

Prototyping

Architekturreview

Code-Inspektion



Testplan u. Unit-Test-Entw.

Unit-/Integrationstests



Implementierung

Infrastruktur einrichten, Teambuilding

beeinflusst/startet

Initiale Projektplanung

prüft

Ist-Analyse

Soll-Analyse

Usability-Prototyp

Akzeptanztest
(Präsentation)



Anforderungsspezifikation



Review der Anforderungs-
spezifikation



Handbuch
schreiben



Testprotokoll

Architekturentwurf

Systemtests

Technisches
Prototyping



Architekturreview

Code-Inspektion

Testplan u. Unit-Test-Entw.

Implementierung

Unit-/Integrationstests

Infrastruktur einrichten, Teambuilding

Initiale Projektplanung

Ist-Analyse

Soll-Analyse

Usability-Prototyp



Anforderungsspezifikation



Review der Anforderungs-
spezifikation

Handbuch
schreiben



Architekturentwurf

Technisches
Prototyping

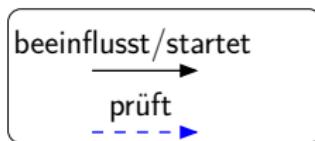


Architekturreview



Testplan u. Unit-Test-Entw.

Implementierung



beeinflusst/startet

prüft

Übergabe

Akzeptanztest
(Präsentation)



Testprotokoll

Systemtests

Code-Inspektion

Unit-/Integrationstests

Scheinbedingungen

[http://www.informatik.uni-bremen.de/st/Lehre/swpI_13/
scheinbedingungen.html](http://www.informatik.uni-bremen.de/st/Lehre/swpI_13/scheinbedingungen.html)

Scheinbedingungen

http://www.informatik.uni-bremen.de/st/Lehre/swpI_13/scheinbedingungen.html

Ausblick Wintersemester 2013/14:

- Teilnahme auch ohne SWP-I-Schein möglich
- **aber:** keine gemischten Gruppen von Teilnehmern mit und ohne SWP-I-Schein

Tutorienwahl

Anmeldung bis zum bis 4.04., 23.59 Uhr² über MEMS:

<http://www.informatik.uni-bremen.de/st/mems/courses.php?lng=de>

- nur in Sechsergruppen anmelden
- nur mit TZI-Email-Adresse anmelden
- Gruppennamen sorgfältig auswählen

²Ortszeit Bremen



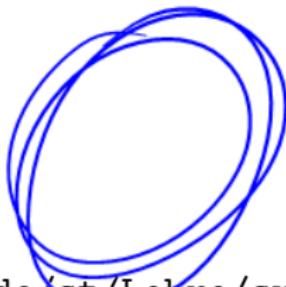


Technische Randbedingungen

- Implementierung in Java 5 oder höher
- Mehrbenutzerbetrieb
- Client/Server-Anwendung
- relationale Datenbank muss für die Persistenz benutzt werden:
Persistenz-Frameworks erlaubt, aber dann deklarative Verwendung von
SQL-ähnlichen Abfragen verlangt
- Plattformen: Windows, Linux, Android



Ressourcen



- Web-Seite zur Vorlesung:
http://www.informatik.uni-bremen.de/st/Lehre/swpI_13/
- Folien mit Kommentaren und Folien mit Annotationen aus der Vorlesung bei Stud.IP:
<http://elearning.uni-bremen.de/>
→ dort registrieren
- Video-Aufzeichnung aus dem letzten Jahr:
<http://mlecture.uni-bremen.de>